



Fall 2023

CSCI-41

**Programming Concepts and Methodology II
Syllabus**

Instructor: Simon Sultana, Ph.D.

Department: Computer Science

E-mail: simon.sultana@reedleycollege.edu

Phone: (559) 638-0300 x3192

Response time: < 24 hours

Office Hours: Mon, Wed 11am-12pm, 3-4pm
(PHY-81), or by appt

Prerequisites: CSCI-40 or ENGR-40.

Section Number: 51063

Unit(s): 4

Dates: 8/7/2023 – 12/11/2023

Class Meeting: Mon & Wed 9-10:50am

Location: PHY-82

Advisory: ENGL-125 – Writing Skills for College and ENGL-126 Reading Skills for College

Course Description: This course introduces application of software engineering techniques to the design and development of large programs, data abstraction and structures, and associated algorithms. Topics include linear and non-linear data structures such as lists, stacks, queues, trees, and graphs, algorithms for recursion, searching, sorting, and traversal.

Note: this course is offered in a hybrid format. There will be a recorded lecture available at the beginning of the week. Students are expected to work on zyBooks activities before our second weekly class meeting. The meeting will be used to work on class activities in small teams. The meetings are offered in two-way interactive format, meaning that students have the choice of joining on campus or via Zoom. Please note that live attendance is required each week (in person or online). It is up to each student how they would like to attend each week. Note we will be working on labs in small groups and we will combine students who are in person and online. Note also that students are required to attend the quizzes (Wednesday, September 6; Monday, October 2; and Wednesday, October 25) and the final exam (Monday, December 4, 2023) on campus. Please let me know if you have any questions.

Course Goals and Student Learning Outcomes:

Upon completion of this course, students will be able to:

- 🔥 Develop computer programs to meet given requirements.
- 🔥 Utilize classic data structures; including linked lists, stacks, queues, trees, and graphs in computer programs.
- 🔥 Analyze the efficiency of algorithms.
- 🔥 Write programs using advanced programming concepts; including constructor and operator overloading, inheritance, abstract classes, and templates.

Objectives:

In the process of completing this course, students will:

- 🔥 Utilize binary search tree data structure.
- 🔥 Use string, stack and queue classes defined in C++ Standard Libraries.
- 🔥 Identify and correct syntax and logical errors in computer programs.
- 🔥 Utilize binary search tree data structure.

- 🔥 Create proper test cases to test computer programs.
- 🔥 Define and use dynamic arrays, linked list, stacks and queue data structures.
- 🔥 Design, implement, and test the implementation of "is-a" relationships among objects using a class hierarchy and inheritance.
- 🔥 Use string, stack and queue classes defined in C++ Standard Libraries.
- 🔥 Identify and correct syntax and logical errors in computer programs.
- 🔥 Write a total of 1000 to 2000 lines of program code.
- 🔥 Write programs using object-oriented programming and the C++ language.
- 🔥 Write programs using pointers, recursion and file manipulation techniques.
- 🔥 Examine ethical issues in software engineering

Student Learning Outcomes are statements about what the discipline faculty hope you will be able to do at the end of the course. This is NOT a guarantee: the ultimate responsibility for whether you will be able to do these things lies with you, the student. In addition, the assessment of Student Learning Outcomes is done by the department in order to evaluate the program as a whole, and not to evaluate individual faculty performance.

Required or Recommended Textbooks and Materials:

Textbooks:

1. **Required:** zyBooks, register and connect via Canvas (cost: \$58).
2. **Optional:** *Starting out with C++ From Control Structures through Objects*, 9th Ed, By Tony Gaddis, Pearson

Additional Resources:

Towson University Cybersecurity Modules: <https://cis1.towson.edu/~cssecinj/>

Learning Management System: CANVAS:

Canvas (<https://sccd.instructure.com/>) is used to post announcements, course information, programming assignments, and grade. You will submit your programming assignments on Canvas. To log-in Reedley College CANVAS:

Username: Your 7-digit student ID number.

Password: If you have not previously changed your password, it is:

First name initial (upper case) + *last name initial* (lowercase) + *date of birth* (mmddyy)
 Example: John Smith born on July 9th of 1988 Password = Js070988

Computer Lab:

Onsite face-to-face meetings will take place in FEM 3 on campus. Microsoft Visual Studio C++ compiler is used for this class.

Topics:

A. Classes

1. Classes and Members; Friend Functions
2. Constructors and Destructors
3. Using a Namespace, Header file, and Implementation file
4. Operator Overloading

B. Inheritance, Polymorphism, and Virtual Functions

1. Introduction to Inheritance
2. Protected members and Class Access
3. Constructors and Destructors in Base and Derived Classes

4. Redefining Base Class Functions
5. Class Hierarchies
6. Polymorphism and Virtual Member Functions
7. Abstract Base Classes and Pure Virtual Functions

C. Pointers and Dynamic Memory

1. Pointers and arrays as parameters
2. A container class with a dynamic array

D. Linked Lists

1. A fundamental node class for linked lists
2. A linked-list toolkit
3. A container class with a linked list

E. Software Development with Templates, Iterators, and the STL

1. Template functions
2. Template classes
3. Standard template classes and their iterators

F. Stacks

1. Stacks and the STL stack
2. Stack applications

G. Queues

1. Queues and the STL queue
2. Queue applications

H. Recursion

1. Recursive functions
2. Recursive applications

I. Binary Search Trees

1. Binary trees
2. Binary tree representations
3. Binary tree nodes
4. Binary tree traversals
5. Binary search trees

J. Hash Tables

1. Chaining
2. Linear & quadratic programming
3. Hashing

J. Graphs

1. Graph definition
2. Graph implementation
3. Graph traversal
4. Path algorithm

K. Further topics

1. Repositories

2. Programming style
3. Secure programming
4. Engineering ethics
5. Heaps

Tentative Schedule:

	Assignment Points	Date
Week 0: CSCI-40 Topics Review	20	8/7/2023
Class 9am - Intro to Class		8/7/2023
Class 9am		8/9/2023
In-Class Activity	10	8/11/2023
zyBooks Participation Activities (optional for review)	-	8/13/2023
Project #0 Proposal	10	8/14/2023
Week 1: More on Objects & Classes	41	8/14/2023
Class 9am		8/13/2023
Optional Reading: Starting Out with C++ (SOWC++) Ch. 9.8, 9.10, 13.9, 14		8/15/2023
Class 9am		8/15/2023
zyBooks Participation Activities	7	8/18/2023
In-Class Activity	10	8/18/2023
zyBooks Week 1-3 Challenge Activities	14	9/3/2023
Schedule One-on-One	10	8/20/2023
Week 2: Inheritance & Abstract Classes	47	8/21/2023
Optional Reading: SOWC++ Ch. 15		8/23/2023
Class 9am ONLINE ONLY		8/21/2023
zyBooks Participation Activities	7	8/25/2023
Class 9am		8/23/2023
In-Class Activity	10	8/25/2023
zyBooks Challenge Activities	-	9/3/2023
Project #1 - User Interface & Dynamic Array	30	8/28/2023
Week 3: Templates, Style Guidelines, & Intro to GitHub	11	8/28/2023
Class 9am		8/28/2023
Optional Reading: SOWC++ Ch. 16.2-16.4, 17.1-17.5		8/30/2023
zyBooks Participation Activities	1	8/30/2023
Class 9am		8/30/2023
In-Class Activity	10	9/1/2023
zyBooks Challenge Activities	-	9/3/2023
Week 4: Quiz #1 / Stand-Up Meeting	35	9/4/2023
No Class Meeting - Labor Day		9/4/2023
Class 9am - Quiz #1 (covers weeks 0 - 3) ON CAMPUS ONLY – PHY 82	35	9/6/2023
Week 5 : Intro to Data Structures & Algorithms	30	9/11/2023
Class 9am		9/11/2023
zyBooks Participation Activities	6	9/15/2023
Optional Reading: SOWC++ Ch. 11, 17.6		9/13/2023
Class 9am		9/13/2023
In-Class Activity	10	9/15/2023
zyBooks Weeks 5-7 Challenge Activities	14	9/29/2023

Week 6: Searching & Algorithm Analysis		58	9/18/2023
Class 9am			9/18/2023
Optional Reading: SOWC++ Ch. 8.1, 8.2, Read Early Objects Ch. 9.6			9/20/2023
zyBooks Participation Activities	8		9/22/2023
Class 9am			9/20/2023
In-Class Activity	10		9/22/2023
zyBooks Challenge Activities	-		9/29/2023
Project #2 - Array of Structures	40		9/25/2023
Week 7: Sorting Algorithms, Secure Programming #1		22	9/25/2023
Class 9am			9/25/2023
Optional Reading: SOWC++ Ch. 8.3, 8.4, 20.8			9/27/2023
zyBooks Participation Activities	12		9/29/2023
Class 9am			9/27/2023
In-Class Activity	10		9/29/2023
zyBooks Challenge Activities	-		9/29/2023
Week 8: Linked Lists		129	10/2/2023
Class 9am Quiz #2 (on weeks 5 - 7) ON CAMPUS ONLY – PHY 82		34	10/2/2023
Optional Reading: SOWC++ Ch. 18, 20.5			10/4/2023
zyBooks Participation Activities	15		10/6/2023
Class 9am			10/4/2023
In-Class Activity	10		10/6/2023
zyBooks weeks 8-10 Challenge Activities	20		10/20/2023
Project #3 - Searching & Sorting	50		10/9/2023
Week 9: Stacks & Queues		20	10/9/2023
Class 9am			10/9/2023
Optional Reading: SOWC++ Ch. 19			10/11/2023
zyBooks Participation Activities	10		10/13/2023
Class 9am			10/11/2023
In-Class Activity	10		10/13/2023
zyBooks Challenge Activities	-		10/20/2023
Week 10: Hash Tables		71	10/16/2023
Class 9am			10/16/2023
Optional Reading: resources (see Canvas)			10/18/2023
zyBooks Participation Activities	11		10/20/2023
Class 9am			10/18/2023
In-Class Activity	10		10/20/2023
zyBooks Challenge Activities	-		10/20/2023
Project #4 - Linked List	50		10/23/2023
Week 11: Software Engineering Ethics		121	10/23/2023
Class 9am - Stand Up Meeting	10		10/23/2023
Discussion Forum - Software Engineering Ethics	15		10/26/2023
Class 9am Quiz #3 (on weeks 8-10) ON CAMPUS ONLY – PHY 82		36	10/25/2023
In-Class Activity	10		10/27/2023
Project #5 - Stack/Queue	50		11/6/2023
Week 12: Trees		38	10/30/2023
Class 9am			10/30/2023

Optional Reading: SOWC++ Ch. 21		11/1/2023
Class 9am		11/1/2023
zyBooks Participation Activities	13	11/3/2023
In-Class Activity	10	11/3/2023
zyBooks weeks 12-16 Challenge Activities	15	12/1/2023
Week 13: Balanced Trees & Heaps, Secure Programming #2	73	11/6/2023
Class 9am		11/6/2023
Optional Reading: resources (see Canvas)		11/8/2023
Class 9am		11/8/2023
zyBooks Participation Activities	13	11/10/2023
In-Class Activity - Secure Programming	10	11/10/2023
Project #6 - Hash Table	50	11/20/2023
Week 14: Code Review	135	11/13/2023
Class 9am		11/13/2023
Optional Reading: SOWC++ Ch. 20 (skim)		11/15/2023
Class 9am		11/15/2023
Code Review	30	11/17/2023
Project #7 - Final	100	11/30/2023
Project Peer Evaluation	5	11/30/2023
Week 15: Graphs	21	11/20/2023
Class 9am		11/20/2023
Optional Reading: Malik Ch. 20 or resources (see Canvas)		11/22/2023
zyBooks Participation Activities	11	11/24/2023
Class 9am		11/22/2023
In-Class Activity	10	11/24/2023
Week 16: More Algorithms with Graphs	13	11/27/2023
Class 9am		11/27/2023
Optional Reading: SOWC++ Ch. 20 (skim)		11/29/2023
zyBooks Participation Activities	13	11/29/2023
Class 9am		11/29/2023
In-Class Activity - Spanning Tree, Dijkstra's Algorithm, & More Security	-	11/29/2023
Week 17: Finals Week	115	12/4/2023
Final Exam **9am** ON CAMPUS ONLY - PHY 82	100	12/4/2023
Professionalism	15	
TOTAL	1000	

Subject to Change:

This syllabus and schedule are subject to change. If you are absent from class meeting, it is your responsibility to check on any changes made while you were absent.

Evaluation:

Students will be evaluated on the basis of their performance on various assignments according to the following scale. The instructor reserves the right to adjust scores as it may be required throughout the semester.

Points in the course total 1000 and are distributed as follows:

zyBooks Participation Activities (13)

127

zyBooks Challenge Activities (3)	63
In-Class Activities (13 of 14)	140
Programming Projects (8)	380
Quizzes & Exams (4)	205
Software Engineering Activities/Forums/Code Review	65
One-on-one/Professionalism	20

Final grade is assigned using following scale:

895+ points	A
795 – 894 points	B
695 – 794 points	C
595 – 694 points	D
< 595 points	F

Attendance

Attendance for onsite and synchronous online meetings will be taken at beginning of each class. Students who leave before the end of class will be marked tardy. For synchronous online meetings you are expected to have your camera turned on and will have the best experience if you use earbuds/headphones. Please make sure to stay muted until you have a question or something to add so as to cut down on background noise.

Students will be dropped from the class if they fail to attend the first class session of the semester. During the semester up to final drop date, any student who missed two weeks of class meetings (cumulative) will be dropped from this class (i.e. 4 classes).

Make-up tests are limited to students who have made arrangements with the instructor **prior** to the announced testing date or those students who have been excused by High School Attendance Office. Exam material is constructed from class discussions, assigned readings, guest lectures, video presentations, and special assignments. **Unless the student receives prior approval from the instructor, no make-up tests will be allowed.**

Grading Policy

zyBooks Participation Activities:

ZyBooks participation activities are to be completed before announced due date (typically at start of the second weekly class meeting for full credit for augmented credit as shown below, exception in the first two weeks). These activities allow you to actively engage each week's content. Points will be attributed according to the number of responses required and the percentage of the activities a student completes before class.

Percentage Completed	Before Due Date Points Awarded	By End of Week Points Awarded
90%+	100% of points	90% of points
80%+	90% of points	80% of points
70%+	80% of points	70% of points
60%+	70% of points	60% of points
50%+	60% of points	50% of points
25%+	40% of points	30% of points
13%+	20% of points	10% of points

In-Class Activities:

In-class activities will be completed in groups of 2-3 during class meeting time with some outside of class time, as necessary. Only one assignment needs to be submitted per group. Be sure to include names of all students in

the group or credit will not be given. The lowest score on the activities of the following weeks will be dropped: 0, 1, 3, 15, 16.

Programming Assignments:

Seven programming mini-projects are assigned in this course. Students will work individually (Project 0 pre-work, Project 2, Project 5, and Project 6) and in pairs if desired. Students should NOT share code with other individuals or groups unless working in a pair on a group project. Projects are suggested to be turned in for grading by the due date indicated on the course calendar. These are to be done. However, these can be turned in up to two weeks later at no penalty (with the exception of Project #7 as all assignments are due before the week before the final exam) **as long as the student submits a stand-up video on the original due date (see late policy) and the following week, if needing a second extra week.** Stand-ups are required from each student, even in group projects if not meeting the original due date. Please be careful as it is not a good idea to get too far behind and the work will build up and you will likely have a difficult time succeeding in the class. Consideration is given to the following criteria: functionality, error-proofing/exception handling, efficient use of course concepts, documentation, and readability in addition to test case results. All programming exercises must include your name at the top of all programs submitted, or you will not receive credit.

Stand-Up Meetings and Code Review:

A stand-up meeting will be held during class in the week of Quiz #3. Students are to present 2-3 minutes (either in person or online with camera on) on their progress on the current project, what is planned to be accomplished in the next week, and most importantly, their biggest hurdle. The goal is for students to share best practices and get assistance as needed. For the Code Review, students are to present a piece of code they have written and discuss their design and coding choices. These must be done in person or online with a camera on. Students should share their code as they discuss it. Students are also responsible for entertaining questions at the end of their presentation and are responsible for asking questions or making meaningful suggestions to at least two other students. See Canvas for details.

Exams:

There are three quizzes and one final exam. All will be timed. See course calendar for timing, content covered, and points allotted. Tests will consist of multiple-choice questions on syntax, program flow, etc. There will also be small programming problems. A more difficult late exam can only be arranged if you have an excuse verified by an impartial party (i.e., a doctor or a court note).

Late Policy

Points will be minimally deducted for late homework problem sets/zyBooks challenge activities at 0.5 – 1 point for assignments late. Problem sets are to be submitted electronically in Canvas and all work must be shown for partial credit. Programming assignments can be turned in up to **TWO WEEKS** late after the original announced due date, and never later than the announced deadline of project #7. **If you will turn in a project in late, you must submit a stand-up video by the original due date in each week it is late (for each week late)** and you must address the following questions in the video:

- a) What you have accomplished so far (show/share code) focused on the project theme
- b) What you plan to accomplish in the next week
- c) Your biggest hurdle

The video should be 2-4 minutes and can be linked in the comments section of the assignment or shared in the submission area. The focus of the video should be on the specific topic of the project (e.g. linked list, stack or queue, etc.). Failure to meet this deadline will result in a zero on the project.

POGIL exercises will not be accepted late as these are group endeavors.

College Policies:

Cheating & Plagiarism, see Cheating and Plagiarism under Campus Policies:

Cheating and plagiarism is prohibited in the class. Incidents of cheating and plagiarism will result a failing grade on the particular assignment in question. Please see Student Conduct Standards at <https://www.reedleycollege.edu/about/about-us/policies-and-procedures/student%20conduct%20standards.html> for more information about academic integrity.

Each student is expected to assist in the overall environment of the classroom making it conducive to learning.

Code created by external sources can be utilized with permission from the instructor and 1) must be properly documented so as not to plagiarize the work of others and 2) must not make up more than 10% of the actual code (exclusive of documentation) of the project.

It is expected each student will do his/her own work unless otherwise instructed. This course involves both individual work and collaborative work. It is your responsibility to understand the guidelines that apply to each kind of work, and to be clear about which assignments are individual assignments and which are collaborative.

Activities not designated as cooperative assignments in this course are to be done individually. Individual work is to be carried out entirely and solely by an individual. Programs will be checked for plagiarism using both computer and human similarity checkers. Take extreme precautions that your individual work is not viewed by other students. This includes deleting all your computer files from public workstations when you are finished, retaining private permissions on your files, destroying printouts of source code, and not letting other students use your personal computer where you store your coursework.

In addition, the work you submit must be entirely your original creation. Using solutions from any other source is forbidden; in particular, using solutions (either instructors' or other students') from previous offerings of this or other courses is not allowed. Using solutions found on the Internet or getting help from online forums is not allowed.

Assignments which appear to be the result of a "group effort", or appear to have been copied from another student, will be considered plagiarized. Violations of this policy will be reported as violation of academic integrity per college policy.

In-class activities are designated as pair or team activities. Students will be assigned a partner or will self-assign to complete these activities. Be sure to use these collaborative activities as a chance to master the skills, as there will be quizzes in which each person must perform the skills individually.

Accommodations for Students with Disabilities

If you have a verified need for an academic accommodation or materials in alternate media (i.e., Braille, large print, electronic text, etc.) per the Americans with Disabilities Act (ADA) or Section 504 of the Rehabilitation Act, please contact the Reedley College Disabled Students Programs & Services (DSP&S) Department at (559) 638-0332. You can find more information at <https://www.reedleycollege.edu/student-services/disabled-student-programs-and-services/index.html>.

Reedley College is committed to creating accessible learning environments consistent with federal and state law. To obtain academic adjustments or auxiliary aids, students must be registered with the DSP&S office on campus. DSP&S can be reached at (559) 638-3332. If you are already

registered with the DSP&S office, please provide your Notice of Accommodation form as soon as possible.

Important College Dates Fall 2023

Class begin	Monday	08/07/2023
Last day to drop a full-term class for a full refund	Friday	08/18/2023
Last day to register	Friday	08/25/2023
Last day to drop this class to avoid a “W” in person	Friday	08/25/2023
Last day to drop this class to avoid a “W” online	Sunday	08/27/2023
Last date to drop this class	Friday	10/06/2023
No classes, campus is closed		
Veterans Day	Monday	11/10/2023
Thanksgiving Day	Thurs-Friday	11/23-24/2023
Final Exam	Monday	12/04/2023